

## Rapport de Projet de C++

### I. Implémentation Projet

Ce projet a pour but de concevoir et implémenter un mini-jeu sur console. Ce jeu est inspiré de Age of War.

The screenshot shows a terminal window with the following content:

```
maelys@LAPTOP-GDTHCL1K: /mnt/c/Users/mlarribe/Documents/_Cours/ET4/S7/Cpp/AgeOfWar,
Tour de Joueur 2
Joueur 1 91 p
Joueur 2 89 p
100 10 6 3 10 10 10 100
Voulez vous acheter une unite?(f/a/c) (passer:p)
entre:
_
```

The game interface displays the turn for Player 2. It shows the current scores for both players (Joueur 1: 91 p, Joueur 2: 89 p) and the remaining resources (100 for Player 1, 100 for Player 2). The resources are represented by a sequence of characters: 10, 6, 3, 10, 10, 10, 10. The prompt asks the player if they want to buy a unit, with options (f/a/c) and (passer:p). The player has entered a space character.

Mon objectif initial était de produire un code un minimum adaptable. Cette intention se retrouve dans quelques méthodes se basant sur la **généricité** ou les **fonctions anonymes** (`Terrain::add` ou `Terrain::Action3`).

Dans ce projet toutes les unités héritent d'une classe **Unite**. L'aire de jeu quant à elle est représentée par la classe **Terrain**, qui elle-même contient des conteneurs (classe **Case**) de un pointeur d'**Unite** chacun.

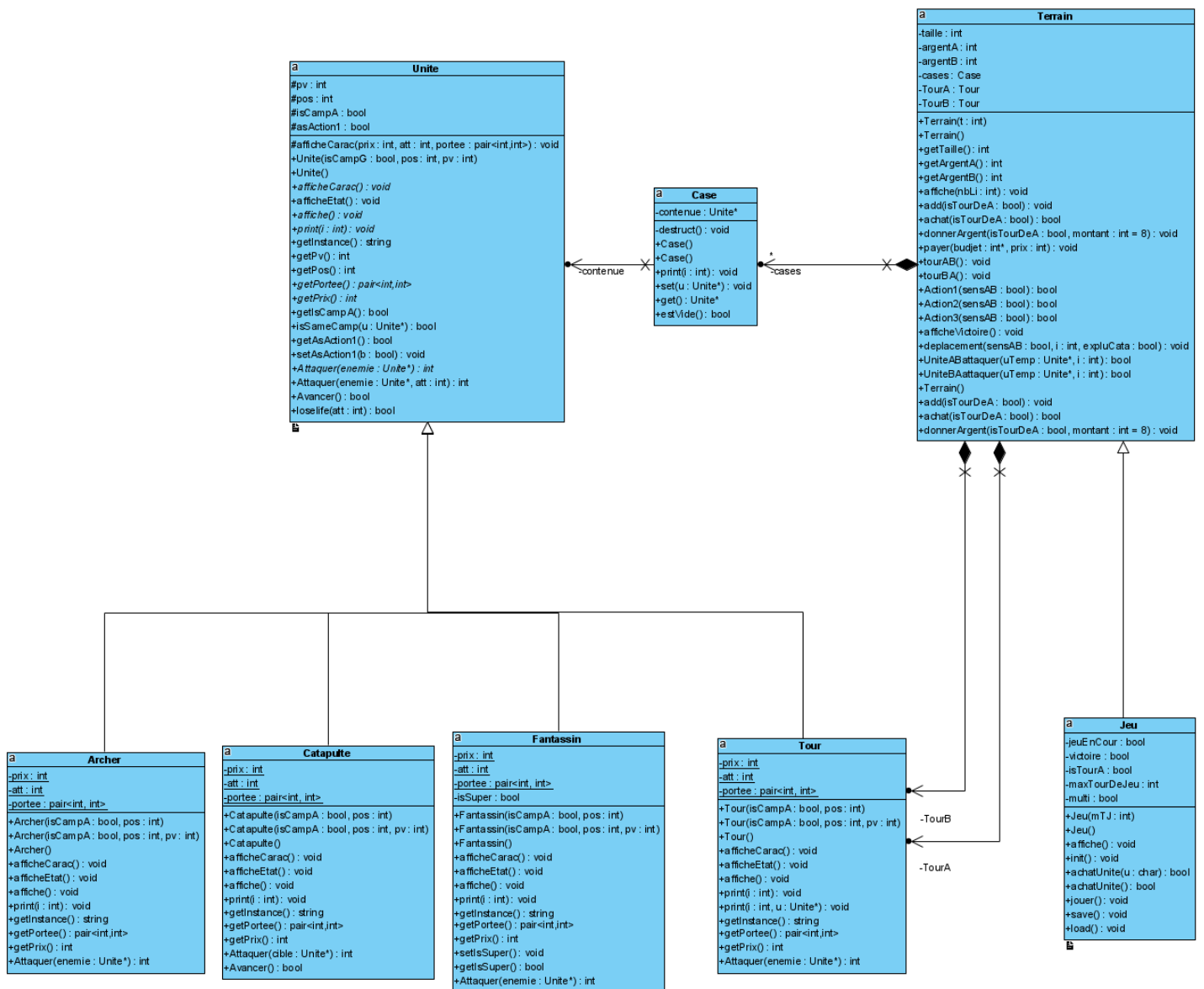
**Terrain** référence également les deux bases.

Le jeu est exécuté avec la méthode `jeu()` de la classe **Jeu**.

L'affichages des unités est fait avec des « bonhommes bâton ».

## II. Diagramme UML

Ci-joint le diagramme d'UML de mon projet à ce jour.



### III. Manuel d'utilisation

Une fois avoir « make » dans le répertoire « /principal », il faut pour lancer le programme écrire « ./main ».

Tel qu'implémenté, « q » est toujours disponible pour quitter le programme.

Au démarrage il est demandé à l'utilisateur s'il souhaite jouer en multijoueur ou contre l'ordinateur. (oui : « y », non : « n »).

Suite à cela, l'utilisateur a l'opportunité de charger une sauvegarde s'il le souhaite (oui : « y », non : « n »).

```
maelys@LAPTOP-GDTHCL1K: /mnt/c/Users/mlarribe/Documents/_Cours/ET4/S7/Cpp/AgeOfWar/principal
```

```
-- Bienvenue dans Age Of War --
Voulez vous jouer en Multijoueur?(y/n)
entre:
y
Voulez vous charger une sauvgarde?(y/n)
entre:
n
```

Une fois la partie lancée, chaque joueur est libre de placer une unité selon son argent disponible (fantassin : « f », archer : « a », catapulte : « c »). Il a également l'opportunité de passer son tour (« p »).

```
maelys@LAPTOP-GDTHCL1K: /mnt/c/Users/mlarribe/Documents/_Cours/ET4/S7/Cpp/AgeOfWar/principal
```

```
Tour de Joueur 2

Joueur 1          51 p
| \ / \ / |
|  o  /  |
|  | - /  |
|  |      |
100 10

   0 \
   o \
   o - o
12

   \ o \ o
   \ | \ |
   \ | \ |
10 10

Joueur 2          100 p
| \ / \ / |
|  o  /  |
|  | - /  |
|  |      |
100

Voulez vous acheter une unite?(f/a/c) (passer:p)
entre:
```

Une fois la partie terminée le joueur a l'occasion de la sauvegarder s'il l'a interrompu (oui : « y », non : « n »).

#### **IV. Difficultés et améliorations**

De nombreuses difficultés se sont présentées lors de ce projet.

##### **Switch case impossible :**

Pour une raison que je n'ai pas réussi à déterminer, je me suis retrouvée dans l'incapacité d'effectuer un switch case.

##### **Mort Tour :**

J'apprécie peu devoir faire une comparaison pour chaque Unite.

##### **Sur vérification :**

Certaines vérifications de pointeur non nul sont effectués à plusieurs endroits dans mon programme, elles ne sont pas nécessairement toutes utiles même si elles protègent le code.

##### **Historique Terrain :**

J'aurais aimé implémenter un système d'historique dans le Terrain qui stock les actions (achat) effectués à un instant t. cela aiderait dans l'isolation de bug.

##### **Super-soldat :**

Pour une adaptabilité plus simple du code, j'aurais préféré prendre le temps de faire une classe à part pour super-soldat.